



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/814,324	03/21/2001	Motohiro Kawahito	JP919990309US1	3779

48233 7590 09/23/2005

SCULLY, SCOTT, MURPHY & PRESSER
400 GARDEN CITY PLAZA
SUITE 300
GARDEN CITY, NY 11530

EXAMINER

NAHAR, QAMRUN

ART UNIT PAPER NUMBER

2191

DATE MAILED: 09/23/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/814,324

Applicant(s)

KAWAHITO ET AL.

Examiner

Qamrun Nahar

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 07/06/2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 9 and 11 is/are allowed.
- 6) ☒ Claim(s) 1-8, 10 and 12-18 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

10

DETAILED ACTION

1. This action is in response to the amendment filed on 07/06/2005.
2. The objections to claims 9 and 11 are withdrawn in view of applicant's amendment.
3. The rejection under 35 U.S.C. 102(e) as being anticipated by Dunn (U.S. 6,247,172) to claims 1-2, 4, 6-7, 10 and 12-16 is moot in view of the new ground(s) of rejection.
4. The rejection under 35 U.S.C. 103(a) as being unpatentable over Dunn (U.S. 6,247,172) in view of Ghosh (U.S. 6,412,109) to claims 3, 5, 8, and 17-18 is moot in view of the new ground(s) of rejection.
5. Claims 1, 6-13 and 15-16 have been amended.
6. Claims 1-18 are pending.
7. Claim 8 stand finally objected to because of informalities.
8. Claims 1-2, 4, 6-7, 10 and 12-16 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Dunn (U.S. 6,247,172) in view of Click, Jr. (U.S. 6,363,522) (hereinafter, Click).
9. Claims 3, 5, 8, and 17-18 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Dunn (U.S. 6,247,172) in view of Click, Jr. (U.S. 6,363,522) (hereinafter, Click), and further in view of Ghosh (U.S. 6,412,109).
10. Claims 9 and 11 are allowed (See the previous Office Action, Mailed on 04/06/2005, par. 13 and 14).

Response to Amendment

Claim Objections

11. Claim 8 is objected to because of the following informalities: “a commands” on lines 3-4 of the claim should be “commands”. Appropriate correction is required.

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

13. Claims 1-2, 4, 6-7, 10 and 12-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dunn (U.S. 6,247,172) in view of Click, Jr. (U.S. 6,363,522) (hereinafter, Click).

Per Claim 1 (Amended):

Dunn teaches a compiler for converting source code for a program written in a programming language into object code in a machine language for a target machine (“FIG. 7 is a simplified block diagram of a compiler embodiment of the translation system according to the present invention. This is another embodiment of the translation system described in FIG. 4. A compiler can be thought of as a translator, translating a source program into compiled code. Compilers can aggressively optimize to produce the most efficient code possible.” in column 7, lines 20-30; and see Figure 7; This *embodiment* teaches a compiler, where the source code is

Art Unit: 2191

translated into object code in a machine language for a *target machine*); an optimization execution unit for performing an optimization process for an object program written in a machine language for a target machine (“The compilation system 70 includes a compiler 74 that converts a source program 72 written in a high order language, such as C, into compiled code 78 operable on the target platform comprising the target hardware and the target operating system 84. The compiler 74 creates compiled code 78, including the optimized compiled code 80 as well as recovery blocks 82. The compiler produces a set of recovery blocks 82, up to one for each potentially trapping instruction in the optimized compiled code 80.” in column 7, lines 31-39); and a program modification unit for modifying said object program in order to absorb differences in content between the points of origin of exception processes, including hardware exception processes, which occur in response to the execution on the target machine of commands in said object program, and an exception handler whereat said exception processes are performed (“When the optimized compiled code 80 *executes on the target operating system 84*, a synchronous exception may result. If a synchronous exception occurs, the target operating system 84 invokes a runtime recovery procedure 86, linked to the application as part of the runtime library. The runtime recovery procedure then *restores the target machine state* using the recovery blocks 82.” in column 7, lines 40-46 (emphasis added); the recovery block compensates for the difference in content from the point where the exception occurs and the exception handler. See also column 6, lines 1-18, *where Dunn teaches compensating a register image between the point of origin of the exception occurrence points and the exception handler.*).

Dunn does not explicitly teach software exception processes. Click teaches software exception processes (column 6, lines 13-16).

Art Unit: 2191

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the compiler disclosed by Dunn to include software exception processes using the teaching of Click. The modification would be obvious because one of ordinary skill in the art would be motivated to handle exceptions as part of normal program control flow (Click, column 3, lines 10-14).

Per Claim 2:

The rejection of claim 1 is incorporated, and Dunn further teaches wherein, if there is a difference in content between the point of origin of an exception process, which occurs in response to the execution of a command in said object program, and a location whereat said exception process is performed, said program modification unit generates compensation code to compensate for said difference, and inserts said compensation code into said object program (column 7, lines 35-46).

Per Claim 4:

The rejection of claim 1 is incorporated, and Dunn further teaches wherein, before said optimization execution unit performs said optimization process in said object program, said program modification unit divides said command that may cause an exception process into a command portion for determining whether an exception process has occurred, and a command portion for actually causing an exception process; and wherein, when an exception process occurs, said program modification unit modifies said object program to shift program control to said command portion that actually caused said exception process (column 6, lines 9-32).

Per Claim 6 (Amended):

This is a system version of the claimed compiler discussed above, claim 1, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above.

Thus, accordingly, this claim is also obvious.

Per Claim 7 (Amended):

This is a method version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above, including “moving program control to an exception handler whereat said exception processes are performed” (Dunn, column 6, lines 10-15). Thus, accordingly, this claim is also obvious.

Per Claim 10 (Amended):

Dunn teaches an optimization method for optimizing a program to increase processing efficiency on a target machine (column 7, lines 20-30), comprising the steps of: dividing software code, in an object program, that may cause exception processes, including hardware exception processes on the target machine into software code for determining whether an exception process has occurred and software code for actually causing an exception process (column 7, lines 31-39; and column 6, lines 36-67 to column 7, lines 1-6); specifying said code obtained at said division step as branches of a control flow graph; designing said control flow graph so that when one of said exception processes occur on the target machine, program control

Art Unit: 2191

is shifted to said code that actually caused said one of said exception processes; and performing said optimization process for said object program that has been modified (column 7, lines 40-46; column 6, lines 36-67 to column 7, lines 1-6; and see Figure 6). Dunn does not explicitly teach software exception processes. Click teaches software exception processes (column 6, lines 13-16).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Dunn to include software exception processes using the teaching of Click. The modification would be obvious because one of ordinary skill in the art would be motivated to handle exceptions as part of normal program control flow (Click, column 3, lines 10-14).

Per Claim 12 (Amended):

This is a computer program version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 13 (Amended):

This is a computer executable program version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 14:

Art Unit: 2191

The rejection of claim 13 is incorporated, and Dunn further teaches wherein said function for determining whether an exception process has occurred is provided by a condition branch (column 6, lines 56-67 to column 7, lines 1-6).

Per Claim 15 (Amended):

This is a storage medium version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 16 (Amended):

This is a apparatus version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

14. Claims 3, 5, 8, and 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dunn (U.S. 6,247,172) in view of Click, Jr. (U.S. 6,363,522) (hereinafter, Click), and further in view of Ghosh (U.S. 6,412,109).

Per Claim 3:

The rejection of claim 1 is incorporated, and Dunn further teaches wherein said program modification unit includes: a pre-processor for, before said optimization execution unit performs said optimization process, examine a command that may cause an exception process in said

Art Unit: 2191

object program to determine whether an exception process has occurred, and performing an inherent process when it is found an exception process has occurred (column 7, lines 31-39), and a post-processor for examining, in said object program that has been optimized by said optimization execution unit, said command that may cause an exception process to determine whether a difference in content exists between said command that may cause said exception process and a location whereat said exception process is performed, and for, when a difference exists, generating a compensation code, to be used to compensate for said difference, and a code for, after said compensation code is obtained, moving program control to an exception handler whereat said exception process is performed (column 7, lines 40-46 and column 6, lines 19-35). The combination of Dunn and Click does not explicitly teach try-catch blocks for handling exceptions. Ghosh teaches try-catch blocks for handling exceptions (column 5, lines 26-34).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the compiler disclosed by the combination of Dunn and Click to include try-catch blocks for handling exceptions using the teaching of Ghosh. The modification would be obvious because one of ordinary skill in the art would be motivated to increase the amount of code that can be optimized, even code that is associated with a try-catch block (Ghosh, column 6, lines 51-60).

Per Claim 5:

This is another version of the claimed compiler discussed above (claims 3 and 4), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 8 (Amended):

This is a method version of the claimed compiler discussed above (claims 1 and 3), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 17:

The rejection of claim 3 is incorporated, and Dunn further teaches wherein the post-processor registers an address of the point at which the exception process occurs (column 6, lines 19-35).

Per Claim 18:

The rejection of claim 17 is incorporated, and Dunn further teaches wherein the post-processor registers an address of the exception handler (column 6, lines 19-35).

Allowable Subject Matter

15. Claims 9 and 11 are allowed (See the previous Office Action, Mailed on 04/06/2005, par. 13 and 14).

Response to Arguments

16. Applicant's arguments with respect to claims 1-18 have been considered but are moot in view of the new ground(s) of rejection.

In the remarks, the applicant argues that:

a) Also, the preferred compensation code of the present invention “to absorb a difference ...” has a different objective from Dunn’s recovery block 42. The target machine state of compensation code ... For example, the present invention may be used to perform algorithms in Figs 4 to 7 in order to compensate a register image between the point of origin of the exception occurrence points and the exception handler. To take an example in Fig. 10 of the present invention, the Dunn, et al., approach would not generate “copy i and j variable values to R1 and R2” in the recovery block.

Examiner’s response:

a) Dunn is relied upon for the limitation “to absorb a difference ...”. Dunn teaches a program modification unit for modifying said object program in order to absorb a difference in content between the point of origin of an exception process, which occurs in response to the execution on the target machine of a command in said object program, and an exception handler whereat said exception process is performed (“When the optimized compiled code 80 *executes on the target operating system 84*, a synchronous exception may result. If a synchronous exception occurs, the target operating system 84 invokes a runtime recovery procedure 86, linked to the application as part of the runtime library. The runtime recovery procedure then *restores the target machine state* using the recovery blocks 82.” in column 7, lines 40-46 (emphasis added) and see Figure 7; the recovery block compensates for the difference in content from the point where the exception occurs and the exception handler. See also column 6, lines 1-18,

where Dunn teaches compensating a register image between the point of origin of the exception occurrence points and the exception handler.).

In addition, see the rejection above in paragraph 13 for rejection to claims 1-2, 4, 6-7, 10 and 12-16.

In the remarks, the applicant argues that:

b) With particular regard to Claim 4, the Dunn, et al. approach does not disclose claim 4. Claim 4 describes that an application program (or software) detects and throws an exception. The Dunn, et al. approach (Dunn's column 6, lines 1-35) supposes that the hardware detects and throws an exception. The compiler of Claim 4 divides a software exception check into a detection portion and a throwing exception portion within "the application program" as shown in Fig. 26. Therefore, this approach enables that an exception is directly transferred within the application program. In contrast, Dunn's approach is limited by the fact that an execution must be transferred via OS (Dunn's column 6, lines 1-35).

Examiner's response:

b) Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dunn in view of Click. Dunn is relied upon for the limitation of claim 4. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., software exception check) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26

Art Unit: 2191

USPQ2d 1057 (Fed. Cir. 1993). That is, claim 4 does not recite software exception check.

Therefore, hardware exception processes meet the claim limitation.

In addition, see the rejection above in paragraph 13 for rejection to claim 4.

Conclusion

17. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

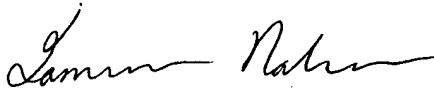
18. Any inquiry concerning this communication from the examiner should be directed to Qamrun Nahar whose telephone number is (571) 272-3730. The examiner can normally be reached on Mondays through Fridays from 9:30 AM to 6:00 PM.

Art Unit: 2191

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or processing is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



QN
September 15, 2005



TUAN DAM
SUPERVISORY PATENT EXAMINER